

PROGRAMMABLE KEYBOARD is still up in the air with nothing new to report in that area. I have received some positive responses to my query on projects in work, having to do with adding a keyboard and with adding memory - if there are any more workers out there, please let me know. I am now thinking that it would probably be a good idea to consider a backup plan in case the Bally output is delayed even more. This plan would provide us with a full size keyboard and added memory, but without the GRAFIX or ZGRASS features. This would come about by the concentrated effort of a number of subscribers who currently are working independently on either a keyboard or memory addition. With a common goal in mind, coordination of effort among the individuals, and probably a deadline (Jan 80), we should be able to progress faster than we are now. One important question is cost, and I am going to ask for a show of hands on the following which will enable us to determine the amount of memory capability that can be incorporated plus other 'goodies'.

1. Assume that the Bally Keyboard is available with full capacity (ref. p. 21).
Are you ready to pay 650. for it?
2. Assume that the Bally Keyboard is available with partial capacity (ref. p. 54).
Are you ready to pay 350. for it?
3. Assume that we develop a keyboard that would have 16K RAM with upgrading capability to 24+ RAM, and some form of resident BASIC in 16K ROM, along with some features as cassette motor control, word processing capability, etc.,
Are you ready to pay 350. for it? (Assuming that Bally does not produce in the same time frame.)

A postal card with numbers down the side and yes/no opposite each is all that is necessary, but suggestions are certainly welcome. Also, tell me Model Number and Serial Number of your machine if you haven't done so yet.

INPUT DOUBLING has been reported by Kirk Gregg. He notes that you can include as many as 36 variables in an INPUT statement, separated by commas. That way you can INPUT A,B,O,U, @(9),@(6),etc., saving a few bytes.

DIRECT OPERATION of equipment is one of my pet goals for this machine. Some time back Mitt Nodacker sent me a scheme for detecting a specific tone output of the tone generator using a phase lock loop (PLL) decoder. This device listens for a tone of specific frequency and creates an output when the tone is detected. The output can then drive another device. The circuits are covered in such documents as: FM AND REPEATERS, by ARRL, p.120; INTEGRATED CIRCUIT PROJECTS Vol 3 by Radio Shack, p.57; or INTEGRATED CIRCUIT PROJECTS by Sams, ch.3., all using the NE567 PLL device and a few resistors/capacitors. I tried the circuit without any success, but a letter from Rich DeLong and Dan Zielinski stated that a frequency counter is most useful, so maybe I was just out of adjustment. But along with their letter came a similar circuit with a two-step amplifier and relay with which they have had success. I'll work on this and give you a report next time.

TRUTH of a statement can be directly displayed by the use of the PRINT command, reports Marc Gladstein. This is an adjunct to the IF discussions of pp. 52 + 53, where the machine decides if a statement is true (1) or false (0). The answer can be displayed, if you are interested, as follows:

| | | |
|-------------|---------|---|
| PRINT 1 = 2 | display | 0 |
| PRINT 3 = 3 | display | 1 |
| PRINT 1 < 2 | display | 1 |
| PRINT 5 > 6 | display | 0 |

arcadian

TUTORIAL #1. Some material for the beginner in programming and computer usage.

First off, the machine is ignorant, totally ignorant with no imagination. It works only when you tell it to, and it only does what you tell it to do. It will do nothing by itself, or arbitrarily.

Everything that you tell it to do must be done in a step-by-step fashion. There are a couple of short-cuts available, but in general you must state everything that must be done and in the order in which it must be done.

As an example of the machine's 'dumbness', it won't do anything if you key in a command on the keypad. All that happens is that the screen will show the keys that were pressed. The information is stored in a space called a "buffer", and you have the opportunity to look over the entry and make corrections if needed. When you are happy with it, press GO, which moves the entry from the buffer into the machine's memory and/or executes it. (Execute meaning 'perform')

To create some sort of order, we generate a 'program' which is a total list of orders to the machine to tell it what to do, plus the permanent data it needs for the solution, and provisions for entering data of a variable sort. To get to the program, one first starts with the step of 'naming the problem'. Then one makes up a 'flow chart' to get the general sequence of operations, and then finally the detailed program. When the program is printed (either on the screen or by hand, etc.,) it is 'listed', or you make a 'listing'. Here is an example of the above in the addition of numbers, five in this case:

PROBLEM FLOW CHART

ADD FIVE NUMBERS

INPUT NUMBERS

ADD NUMBERS

PRINT ANSWER

PROGRAM

```
INPUT A      GO
INPUT B      GO
INPUT C      GO
INPUT D      GO
INPUT E      GO
F = A + B + C + D + E      GO
PRINT F      GO
```

(From now on, I won't print GO for every line entry, it should be a reflex.)

With the program shown above, the problem will work only once. To keep the program in the machine's memory, it is necessary to prefix each line with a 'line number'. These are usually shown in increments of 10, as follows:

```
10 INPUT A
20 INPUT B
30 INPUT C
40 INPUT D
50 INPUT E
60 F = A + B + C + D + E
70 PRINT F
```

Now, to make it operate, you command RUN and then press GO which in this case makes it execute. The computer looks for the lowest line number, reads the instruction there, and reacts. When it has finished reacting, it looks for the next line number, reacts, etc., etc.

You have probably noted that it does not make any difference in what order you enter line numbers, the machine will automatically put them in numerical order, a blessing.

In our example, line 10 says INPUT A so the internals of the machine, constructed by the engineer, decide that the keypad will provide a numerical value, and it will be identified as 'A' and placed in a memory slot, or "register" entitled A.

It tells you this by printing the letter A on the screen followed by the cursor, indicating that it is waiting for you to pump in the first number. When you do, and follow it with GO, the machine has then completed its line 10 task and moves on to line 20. And so forth. At line 60 it sees that it has to set up a new register, F, and it must sequentially add the contents of registers A through E together, and place them in F. This job completed, it looks at line 70 and sees that it has to print the contents of F, which it does on the screen.

While the above is acceptable for a small number of input values, the program length (and therefore memory) can be prohibitive if you had the 26 maximum values you could enter (letter A - Z). In such instances, the FOR-TO loop is useful. We can also take advantage of being able to modify a register. The program is:

```

10 A = 0
20 FOR B = 1 TO 5
30 INPUT C
40 A = A + C
50 NEXT B
60 PRINT A

```

(note that there are only 6 lines to the whole program.)

Analysis:

Line 10 is not really a part of the program, its intent is to make sure that there is no 'left-over' from some other program residing in A when you start. It is called "initialization".

Line 20 starts the FOR - TO loop, and eventually needs line 50 to make the loop work. The machine reads that you will make 5 entries. They will be numbered 1,2,3,4,5. It will not work if you have less than 5 entries (you may have to enter zeros to make up 5), and it will not accept more than 5. The numbering is arranged through a function called "STEP". If you don't define a value of STEP, the machine automatically picks "1". So, the value of B will start at 1 and automatically STEP by ones up to the value of 5.

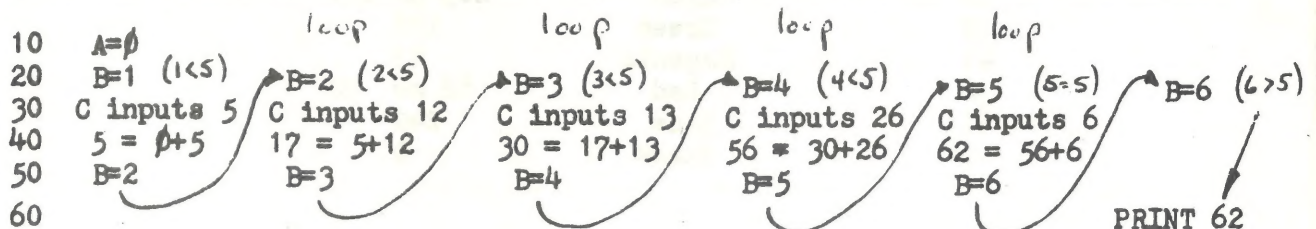
Line 30 asks for inputs as before, but will always assign the input to register C (and wipe out what was there before).

Line 40 operates on register A in the modification mode as was mentioned above. What this says in English is that the New value in the A register is equal to the Old value in the A register plus the value in the C register. In the beginning of the program, the Old value in the A register is zero (Line 10). You now input C and in line 40 it calculates $C + 0$ so that now the New value of A is numerically C.

Line 50 completes the FOR-TO loop instruction. Now is the time for the value of B to change per the STEP function. The machine then goes on to line 30 again, and again, etc., until it runs out of B values, when it goes out of the loop and on to line 60.

Line 60 tells it to print whatever is in register A on the screen.

Lets add up the following numbers: 5, 12, 13, 26, 6 in an extended example:



While the FOR-TO loop saves space, it does so at the expense of losing the identities of the inputted numbers. With the longer system, you could always call up a value, for example: PRINT D and the fourth value would show up. So each has its usage.

Suppose we want to make a small addition to the program, adding one line

Note that since we have used line numbers with an interval of 10, it is easy to slip the addition into the program. The keypad simplifies the 10 interval by having the +10 on the GO key - when you have finished a line, press WORDS +10GO and there it is.

BANGMAN change was suggested by Rory Wohl as line 2000 was giving him some fits:
2000 E=E+1;IF E=9 GOTO 9000; IF Q# 1 GOSUB 9600 +(Ex10)

aMAZed in SPACE game included in this issue is a rocketship-thru-the-maze challenge with a number of levels of difficulty. One problem is that I've lost the name of the originator. I sent the material to Dick Hauser who made a few modifications and prepared the descriptive material. Note how he has separated the listing into blocks that correspond with the flow chart. The program lines marked C are just for information and do not go into the machine.

PROGRAMMING SHEETS that I use in these issues are available from Chuck Thomka 1228 W 222 St. Torrance, CA 90502 at 20 for \$1, add'l sheets at .05 each. He also has the GRAPHICS GRID per the sample shown, for those of you who need an accurate layout of the screen for your graphic displays, same price.

[illegible]

SPECIAL EFFECTS that you may come up with, either visual or aural, are requested. These would be short programs that give you a 'shot' sound, or a 'starburst', or similar enhancement that could be used in a game, etc. I would also keep these in a separate document, and probably make it available later, with an index.

TELEPHONE DATA COUPLER article in KILOBAUD, June '79 has been tried by B.Reany and found to be a workable unit. It uses a Radio Shack 43-230 telephone amplifier and a few parts

COLOR values requested in the last issue brought forth the following response from Don Wurst. "I used a Tektronix 520A vectorscope with a standard TV demodulator and found the following BC's to be as close as possible to the standards."

| <u>Luminance value Y</u> | <u>Color</u> | <u>BC</u> |
|--------------------------|--------------|------------|
| 100 | White | 7 |
| 89 | Yellow | 126 or 134 |
| 70 | Cyan | 205 or 221 |
| 59 | Green | 172 |
| 41 | Magenta | 43 |
| 30 | Red | 82 or 90 |
| 11 | Blue | 249 |
| 0 | Black | 0 |

5 .SLOT MACHINE
6 .BY ERNIE SAMS
7 .4/1979
8 .

```
10 CLEAR:NT=1
15 GOSUB 3000
20 BOX -7,-2,134,64,1
30 BOX 61,0,6,4,1
40 BOX 65,10,3,24,1
50 BOX 67,22,6,10,1
52 BOX -7,-20,126,14,2
54 BOX -7,-20,124,12,1
56 BOX -7,-20,122,10,2
60 CX=-62;CY=-20
70 PRINT "ARCADE SLOT MACHINE",
80 BOX -50,15,42,14,2
82 BOX -50,15,40,12,1
84 BOX -50,15,38,10,2
86 BOX -8,15,42,14,2
88 BOX -8,15,40,12,1
90 BOX -8,15,38,10,2
92 BOX 34,15,42,14,2
94 BOX 34,15,40,12,1
96 BOX 34,15,38,10,2
100 M=0
200 N=KN(1)+128
210 N=N-20; IF N>10 N=10
212 IF N<1 N=1
220 CX=-75;CY=40
230 PRINT "BET?",#3,N
240 IF TR(1)=0 GOTO 200
250 GOSUB 1000
260 GOTO 200
1000 FOR Y=27 TO 8 STEP -1
1010 BOX 67,Y,9,1,2
1015 BOX 67,Y,5,6,10,1
1020 NEXT Y
1030 FOR Y=8 TO 27
1035 BOX 67,Y,4,6,10,1
1040 BOX 65,Y,3,1,1
1045 BOX 69,Y,10,4,2,2
1050 NEXT Y
1060 NT=0
1070 FOR C=0 TO 4
1080 FOR B=0 TO 2
1090 D=RND(5)
1100 IF C=4 & (B)=D
1110 CY=15
1120 IF B=0 CX=-65
1130 IF B=1 CX=-23
1140 IF B=2 CX=19
1145 FOR Z=16 TO 22
1146 &(Z)=79
1147 NEXT Z
```

SLOT MACHINE

```
1150 IF D=1 PRINT "LEMON",
1160 IF D=2 PRINT "CHERRY",
1170 IF D=3 PRINT "ORANGE",
1180 IF D=4 PRINT "BELL",
1190 IF D=5 PRINT "BAR",
1192 FOR Z=22 TO 16 STEP -1;&(Z)=0;NEXT Z
1194 NEXT B
1195 NEXT C
1198 NT=1;L=M
1200 IF &(0)=1 GOSUB 2000;RETURN
1210 IF &(0)=&(1) IF &(0)=&(2) GOSUB 1300;RETURN
1220 IF &(0)=&(1) GOSUB 1400;RETURN
1230 IF &(0)=2 M=M+(2*XN);GOSUB 1500;RETURN
1240 GOSUB 2000;RETURN
1300 IF &(0)=5 M=M+(18*XN);GOSUB 1500;RETURN
1310 IF &(0)=4 M=M+(14*XN);GOSUB 1500;RETURN
1320 IF &(0)=3 M=M+(10*XN);GOSUB 1500;RETURN
1325 IF &(0)=2 M=M+(8*XN);GOSUB 1500;RETURN
1330 GOSUB 2000;RETURN
1400 IF &(0)=4 IF &(2)=5 M=M+(12*XN);GOSUB 1500;RETURN
1410 IF &(0)=3 IF &(2)=5 M=M+(8*XN);GOSUB 1500;RETURN
1420 IF &(0)=2 M=M+(5*XN);GOSUB 1500;RETURN
1430 GOSUB 2000;RETURN
1500 BOX 0,-40,150,8,2
1510 CX=-75;CY=-40
1512 M=N-N
1515 PRINT "YOU WON $", #3,M-L,
1520 GOSUB 2010
1522 FOR V=1 TO N
1525 &(19)=37;&(17)=31;&(21)=47;&(22)=31
1530 FC=83
1540 FOR U=30 TO 20 STEP -1;&(16)=U;NEXT U
1550 FC=0
1570 FOR U=20 TO 30;&(16)=0;NEXT U
1610 NEXT V
1620 FOR Z=22 TO 16 STEP -1;&(Z)=0;NEXT Z
1630 RETURN
2000 BOX 0,-40,150,8,2
2005 M=M-N;CX=-75;CY=-40;PRINT "SORRY",
2010 CX=5;PRINT "TOTAL $", #3,M;RETURN
3000 PRINT "ARCADE SLOT MACHINE";PRINT
3010 PRINT "KNOB #1 LETS YOU BET FROM $1 TO $10. PULL THE TRIG-"
3020 PRINT "GER TO PLACE YOUR BET."
3030 PRINT "THERE ARE LOTS OF WINNING COMBINATIONS. A CHERRY IN THE LEFT WINDOW ALWAYS IS"
3040 PRINT "A WINNER; BUT A LEMON IN THE LEFT WINDOW NEVER WINS"
3050 PRINT "PRESS GO TO START PLAY.",
3060 K=KP;IF K=13 CLEAR;RETURN
3070 GOTO 3060
```

SLOT MACHINE DOCUMENTATION:

| | | |
|--------|-------|---|
| 10 - | 96 | Instructions and set up slot machine. |
| 100 - | 100 | If a new game is started make Money equal to 0. |
| 200 - | 240 | Bet from \$1 to \$10 with knob #1. |
| | 250 | Start "play" routine |
| | 260 | Play is complete go back to 200 for another bet. |
| 1000 - | 1050 | Pull slot machine handle routine. |
| | 1070 | Display 5 sets of words. |
| | 1080 | In 3 boxes. |
| | 1100 | Store the number that counts in the proper storage location. |
| 1110 - | 1140 | Print the words in the appropriate position. |
| 1145 - | 1147 | Set up a sound for the slot machine. |
| 1150 - | 1190 | Print the appropriate words generated by the random number in 1090. |
| | 1192 | Reset all noisemakers to 0. |
| | 1198 | L establishes the amount of money won or lost as of the last play. Is used in 1515. |
| | 1200 | If the first word (left window) is "lemon" goto loser routine. |
| | 1210 | If all three words are the same go to three-of-a-kind winner routine at 1300. |
| | 1220 | If the program reached this statement there was neither a lemon in the first position nor three-of-a-kind. So, if there is two-of-a-kind go to possible winner routine at 1400. |
| | 1230 | If none of the above but the word "cherry" is in the left window increment the Money counter by two times the amount of the bet. Go to the winner routine at 1500. |
| | 1240 | None of the above so had to be a loser. Go to loser routine at 2000. |
| | 1300 | Three "bar"s pay 18 times the bet. |
| | 1310 | Three "bell"s pay 14 times the bet. |
| | 1320 | Three "orange"s pay 10 times the bet. |
| | 1325 | Three "cherry"s pay 8 times the bet. |
| | 1400 | Two "bell"s and a "bar" in the right window pay 12 times bet. |
| | 1410 | Two "orange"s and a "bar" in the right window pay 8 times bet. |
| | 1420 | Two "cherry"s and a "bar" in the right window pay 5 times bet. |
| | 1430 | If right position is not a bar go to loser routine. |
| | 1500 | Blank out old message. |
| | 1512 | Reduce current win by amount of bet. |
| | 1515 | Display current win. |
| | 1520 | Display current total Money. |
| | 1522- | |
| | 1610 | Sound siren and flash lights the number of times of the bet. |
| | 1620 | Reset noisemakers to 0. |
| | 2000 | Blank out old message. |
| | 2005 | Reduce current total by amount of bet. |
| | 2010 | Return to 260 for another bet. |

PULL THE TRIG-

THERE ARE LOTS OF WINNING COMBINATIONS. A CHERRY IN THE

LEFT WINDOW ALWAYS IS

A WINNER; BUT A LEMON IN THE LEFT WINDOW NEVER WINS"

PRESS GO TO START PLAY.",

K=KP;IF K=13 CLEAR;RETURN

GOTO 3060

| PROGRAM NAME: AMAZED IN SPACE | | PROGRAM DESCRIPTION: Maneuver spaceship thru maze without crashing into walls. Direction is controlled by joystick 1. Path size, maze height, maze width and degree of difficulty are selected by keyboard input. Score is based on these inputs and time taken to complete maze. It takes quite awhile to complete maze interior, so start small. | | COMMENTS ON PROGRAM: See Flowchart | |
|-------------------------------|-----|--|-----------------------------|------------------------------------|---------------------------|
| 1 | 200 | Line # | Statement(s) | Line # | Statement(s) |
| 2 | 210 | | CLEAR; FOR A=1 TO 100 | 700 | SPACESHIP |
| 3 | 220 | | X=0; Y=0 | 710 | C=0; T=0 |
| 4 | | | BOX RND (160)-80, RND (88)- | 720 | X=-@ (1) -2 |
| 5 | | | 44, 1, 1, 1; NEXT A | 730 | Y=@ (2) - (L+2); M=0; N=0 |
| 6 | | | | 740 | R (23)=255; BC=0 |
| 7 | | | | 750 | D=JX (1); E=JY (1) |
| 8 | | | | 760 | M=M+D; N=N+E |
| 9 | | | | 770 | G=3 |
| 10 | | | | 780 | IF M>G, M=G |
| 11 | | | | 790 | IF M<-G, M=-G |
| 12 | | | | 800 | IF N>G, N=G |
| 13 | | | | 810 | IF N<-G, N=-G |
| 14 | | | | 820 | X=X+M; Y=Y+N |
| 15 | | | | 830 | IF D#0, R (21)=255 |
| 16 | | | | 840 | IF E#0, R (21)=255 |
| 17 | | | | 850 | IF D=0 IF E=0, R (21)=0 |
| 18 | | | | 860 | T=T+1; CY=4; PRINT #4, T |
| 19 | | | | 870 | BOX X, Y, 3, 3 |
| 20 | | | | 880 | BOX X-D, Y-E, 1, 1, 3 |
| 21 | | | | 890 | BOX X-D, Y-E, 1, 1, 3 |
| 22 | | | | 910 | BOX X, Y, 3, 3 |
| 23 | | | | | |
| 24 | | | | | |
| 25 | | | | | |
| 26 | | | | | |
| 27 | | | | | |
| 28 | | | | | |
| 29 | | | | | |
| 30 | | | | | |
| 31 | | | | | |
| 32 | | | | | |
| 33 | | | | | |
| 34 | | | | | |
| 35 | | | | | |
| 36 | | | | | |
| 37 | | | | | |
| 38 | | | | | |
| 39 | | | | | |
| 40 | | | | | |
| 41 | | | | | |
| 42 | | | | | |
| 43 | | | | | |
| 44 | | | | | |
| 45 | | | | | |
| 46 | | | | | |
| 47 | | | | | |
| 48 | | | | | |
| 49 | | | | | |
| 50 | | | | | |
| 51 | | | | | |
| 52 | | | | | |
| 53 | | | | | |
| 54 | | | | | |
| 55 | | | | | |
| 56 | | | | | |
| 57 | | | | | |
| 58 | | | | | |
| 59 | | | | | |
| 60 | | | | | |
| 61 | | | | | |
| 62 | | | | | |
| 63 | | | | | |
| 64 | | | | | |
| 65 | | | | | |
| 66 | | | | | |
| 67 | | | | | |
| 68 | | | | | |
| 69 | | | | | |
| 70 | | | | | |
| 71 | | | | | |
| 72 | | | | | |
| 73 | | | | | |
| 74 | | | | | |
| 75 | | | | | |
| 76 | | | | | |
| 77 | | | | | |
| 78 | | | | | |
| 79 | | | | | |
| 80 | | | | | |
| 81 | | | | | |
| 82 | | | | | |
| 83 | | | | | |
| 84 | | | | | |
| 85 | | | | | |
| 86 | | | | | |
| 87 | | | | | |
| 88 | | | | | |
| 89 | | | | | |
| 90 | | | </ | | |

Line #

Statements

Comments

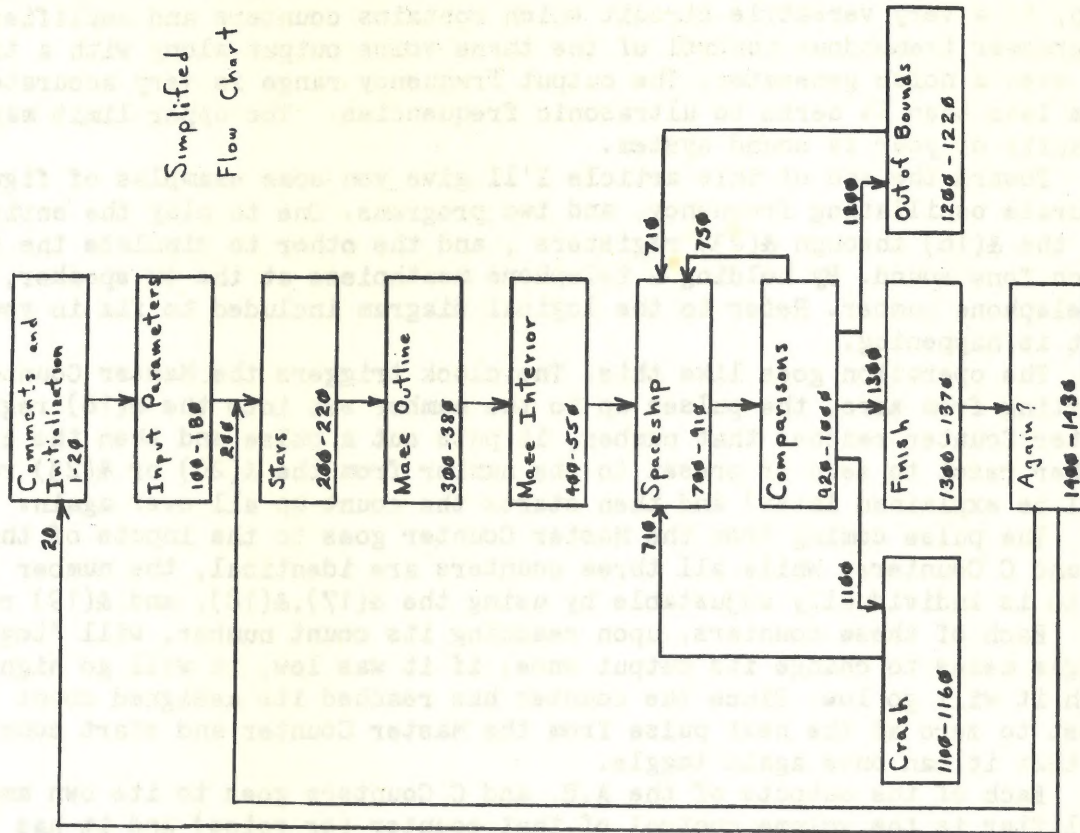
```

1100 CRASH
1110 R(21)=0;C=C+1
1120 CX=-75;CY=44;PRINT C," CRA
1130 SH"
1140 FOR A=1 TO 25;BC=80;NT=5
1150 MU="4";NEXT A
1160 NT=0;GOTO 710
1170
1180
1190
1200 OUT OF BOUNDS
1210 NT=5;CY=44;PRINT "OUTER.LI
1220 MLTS--OFF.LI MLTS";NT=0
1230 CY=44;PRINT "
1240
1250 GOTO 710
1260
1270
1280 FINISH
1290 NT=3;CLEAR
1300 PRINT "FAR OUT! YOU DID IT
1310 !;ONLY";#3,C;" CRASH(CE)1
1320 "
1330 PRINT "TIME=";#5,T
1340 S=((R+1)*H*W)*72/((T*L)/10
1350 )X10
1360 PRINT "SCORE=";#5,S
1370 IF S>J J=S
1380 PRINT "TODAY'S HIGH SCORE="
1390 "#5,J
1400
1410 TO PLAY AGAIN
1420 PRINT "AGAIN?"
1430 NT=0;INPUT "1-YES 2-SAME
1440 AS LAST GAME"Z
1450 IF Z=1 GOTO 200
1460 IF Z=2 GOTO 200
1470 RETURN
1480
1490
1500

```

J- Last Score
 R- Degree of Difficulty
 L- Path Size
 H- Maze Height x2
 W- Maze Width x2
 X- Spaceship X Location
 Y- Spaceship Y Location
 P- # of loops to
 complete maze
 interior, on large
 mazes can take
 15-20 minutes to
 complete
 C- Crashes
 M- X velocity
 N- Y velocity
 D- JX input
 E- JY input
 G- max movement
 per loop, if any
 larger spaceship
 will jump lines
 without crashing
 S- Score
 Z- Again Input
 A, Q loops

Simplified Flow Chart



R.M.H.

TUTORIAL 2 - The Music Synthesizer, by Chuck Thomka

The synthesizer circuit, which is contained wholly within the 40 pin custom I/O chip, is a very versatile circuit which contains counters and amplifiers to give the programmer tremendous control of the three voice output along with a tremolo, vibrato, and even a noise generator. The output frequency range is very accurately adjustable from less than 14 hertz to ultrasonic frequencies. The upper limit may be set by the capacity of your tv sound system.

Toward the end of this article I'll give you some examples of figuring out an accurate oscillating frequency, and two programs. One to play the entire synthesizer via the &(16) through &(23) registers, and the other to simulate the Bell Telephone Touch Tone sound. By holding a telephone mouthpiece at the tv speaker, you can dial a telephone number. Refer to the logical diagram included to fix in your mind just what is happening.

The operation goes like this. The clock triggers the Master Counter which counts, starting from zero, the pulses up to the number set into the &(16) register. When the Master Counter reaches that number, it puts out a pulse and then the counter will either reset to zero or preset to the number from the &(20) or &(23) register (these will be explained later) and then starts the count up all over again.

The pulse coming from the Master Counter goes to the inputs of the identical A, B, and C Counters. While all three counters are identical, the number that each counts up to is individually adjustable by using the &(17), &(18), and &(19) registers.

Each of these counters, upon reaching its count number, will "toggle" its output. Toggle means to change its output once; if it was low, it will go high and if it was high it will go low. Since the counter has reached its assigned count level, it will reset to zero at the next pulse from the Master Counter and start counting up again so that it can once again toggle.

Each of the outputs of the A, B, and C Counters goes to its own amplifier. That amplifier is the volume control of that counter (or voice) and it has 16 levels of output. While the lowest level isn't absolutely zero, it is almost inaudibly low. The &(22) register controls Volume A (the four least significant bits (=LSB)) and Volume B (the four most significant bits (=MSB)) while half of register &(21) (the four LSB) controls Volume C.

There is one more amplifier. This is the Noise Volume. But it must be 'enabled', to work. (that is, turned on) Only one bit is the enable, bit 32 of the &(21) register. This bit must be high for the Noise Generator to be heard at the volume setting of the &(23) register (the four MSB).

&(20) is the Vibrato control register. What it will do to the final audio output is make the sound shift from one frequency to another frequency at a set rate (1 of 4). This makes it very useful for sound effects. How it does this is by not allowing the Master Counter to reset to zero once the Master Counter has reached the &(16) value. Instead, at that time, on the following clock the Master Counter will preset to the value set into the 6 LSB's of the &(20) register. This means instead of the next count cycle of the Master Counter starting at zero, it will start at some number from 0 to 63. This could greatly reduce the time required for the Master Counter to reach its &(16) value. For example, if the &(16) is set to 14, which would normally take 15 clocks (0 to 14 is 15), and the 6 LSB's of the &(20) were set to 10, then this would triple the normal output frequency. This is because now the Master Counter would only have to count from 10 to 14, a total of five clocks, or three times faster than normal.

I mentioned a rate selection of 1 out of 4 choices that the Vibrato will 'vibrate' at. This is selected by the remaining 2 bits of the &(20) register, the 2 MSB's. The four combinations of these 2 bits are 00, 01, 10, and 11, in increasing value. These rates are (approximately) 18.5 milliseconds, 37 milliseconds, 74 milliseconds, and 148 milliseconds, respectively. What happens during this rate time is that for rate 00 the Master Counter will work normally and be allowed to count up to its determined value and then reset to zero for a period of 18.5 milliseconds; then, for the next 18.5 milliseconds the counter will not reset to zero but instead preset to the value of the 6 LSB's in the &(20) register. Of course, if that value is also zero (000000) then there is no difference between reset and preset, hence no vibrato. Each of the other rate selections work in a similar manner except that the duration of normal count time and preset count time will be longer, which is a slower vibrato effect.

My wording earlier was that this present value could greatly reduce the time required for the Master Counter to reach its $\&(16)$ value. It is also possible that the preset value could greatly increase the time, thus making the vibrato shift in audio frequency go lower instead of higher. This is possible when the vibrato value (range 0 to 63) is set to a higher value than the number in the $\&(16)$ register. For example, in my previous example I had $\&(16)$ set to 14 (or 15 counts) and the 6 LSB's of $\&(20)$ set to 10, showing that the resultant vibrato shift would be three times greater than normal. Now in this new example lets have $\&(16)$ still at 14 and put $\&(20)$ (the 6 LSB's) to 15. During the time of vibrato preset operation, when the Master Counter finally gets to 14, its determined value, at the next clock the counter will preset to 15. The counter has no way of knowing that it is now at a count greater than its determined value, it is still looking for the value 14, and 15 is not 14 - so as the counter receives more clock pulses it will continue to count up. And up to count 255 (binary 11111111) it continues. The next clock will overflow the counter to zero (no provisions for carryover are made) and still continue - 1, 2, 3, etc. And it finally reaches count 14 to output the Master Counter Pulse. The next clock repeats the reset to 15 and another long count, "round the horn", to 14. So how many clock pulses does this take? Well, counting 15 to 255 is 241, plus 0 to 14 is an additional 15 for a total of 256. Instead of the normal reset time of 15 clocks, the preset time will take 256 clocks in this example, or over 17 times longer. The longer time for the output pulse from The Master Counter results in a much slower rate of toggling of the A, B, and C counters, which results in a much lower output (voice) frequency.

A Tremolo would be a warble effect in the tone of a note. This is controlled by $\&(23)$, but it must first be enabled by $\&(21)$ bit 16. This bit acts like a switch. When it is off, $\&(20)$ -the Vibrato Register- can be engaged. If $\&(21)$ bit 16 is on, the Vibrato will not work and the Tremolo register is engaged. The Tremolo and Vibrato cannot be engaged at the same time, but as you become more familiar with the sounds out of the Sound Synthesizer, and their ranges, you'll see it is not really necessary anyway.

The operation of the Tremolo is similar to the Vibrato in that it also presets the Master Counter to some number before it starts its count. But, whereas the Vibrato had a rate control (the period of which the counter would alternately reset or preset), the Tremolo always presets. The number to which it is preset is randomly selected and only limited to a maximum number as set into $\&(23)$. All eight bits of $\&(23)$ are used for the Tremolo preset. But remember that the four highest bits are also used for the Noise Volume, in case you try to use the Noise along with the Tremolo. Also remember that the Noise is switched in by $\&(21)$ bit 32.

The preset of a number into the Master Counter, if it is less than the $\&(16)$ determined value of the Master Counter, will shorten the time involved in reaching that $\&(16)$ value. The result will be an upward shift of the resultant A, B, and/or C frequency or pitch, and the next preset will be another randomly selected value (up to the maximum of $\&(23)$). The audible result of all this will be a tremble or quivering effect. If the $\&(23)$ value is greater than the Master Oscillator's $\&(16)$, the resultant may be a random long count 'round the horn' for the same reason that the Vibrato could have a long count. The audible effect will be a wildly varying Tremolo sound. One last remark about the Tremolo is that if $\&(23)$ is set to zero, there will be no Tremolo effect at all since a random preset of up to zero can only be zero, and that is the same as a reset of the Master Counter.

Now lets get into how to figure out a simple set of frequencies. First, the key to the whole thing is to know what the main clock frequency is. I am using a frequency of 1,777,940 hertz for all my figures, measured on pins 16 and 20 of the I/O chip after some warmup. The reciprocal of this frequency is 0.562448 microseconds. This is the unit of time that the Master Counter counts. This should be 1/8 of the crystal oscillator frequency (supposed to be 14,318,180 hertz) but the error in my unit is -.0018%, a quite acceptable figure.

For some practice, here are some parameters:

| | |
|---------------|----------------|
| $\&(16)$ = 60 | Master Counter |
| $\&(17)$ = 99 | Counter A |
| $\&(18)$ = 74 | Counter B |
| $\&(19)$ = 49 | Counter C |

Forgetting for the moment the Vibrato, Tremolo, Noise and Volume controls, I want to show you what the resultant A, B, and C voices would be. First remember that all of these counters start at count zero, which is one less than you or I count on our fingers. Second, since the period of total time it takes for the A, B, or C counter to toggle is only one change of state, it will take yet another toggle to equal one cycle of some resultant frequency. This is why you will find a 2 in the figures below.

Use this formula: (Substitute counter B or C when calculating those frequencies)

$$\text{Freq. A} = \frac{1}{(\text{Master Counter}+1) \times (\text{counter A} + 1) \times \text{Time Unit} \times 2}$$

This converts to:

$$\text{Freq. A} = \frac{1}{61 \times 100 \times 0.562448 \times 10^{-6} \times 2} = 145.7 \text{ hertz}$$

$$\text{Freq. B} = \frac{1}{61 \times 75 \times 0.562448 \times 10^{-6} \times 2} = 194.3 \text{ hertz}$$

$$\text{Freq. C} = \frac{1}{61 \times 50 \times 0.562448 \times 10^{-6} \times 2} = 291.4 \text{ hertz}$$

So there are three different frequencies, all available at one time, by simply turning on the A, B, and C volumes. Notice that while Counter A + 1 is twice the value of Counter C + 1, the resultant frequencies have Freq. C twice the value of Freq. A. This is because Counter A has to count twice as many Master Pulses, before toggling, as Counter C does.

Now all that is useful if you are trying to find the resultant frequency for some known values, but we usually want to go the other way. As an example, we will see the steps to find the values need to generate the Bell System Touch Tones, the number "0". Bell Systems require two frequencies outputted whenever one key is pushed on their regular pad. When pushing "0" the two frequencies are supposed to be 1336 and 941 hertz, with an acceptable tolerance of ± 5 hertz. Calculate Freq. A values for 1336 hertz and Freq. B values for 941 hertz:

Step 1 Find the reciprocal of the frequency

Step 2 Divide the reciprocal by the Time Period

Step 3 Divide by 2, and use the nearest whole number. This answer is the total number of time periods that must be counted before having an output toggle for the frequency's half cycle.

For Freq. A, 1 divided by 1336 divided by 0.562448×10^{-6} divided by 2 = 665.39 = 665

For Freq. B, 1 divided by 941 divided by 0.562448×10^{-6} divided by 2 = 944.7 = 945

The values of 665 and 945 are the total number of clock periods as mentioned above. But recall that the Master Counter must count some of those clocks, and it sends a pulse to both A and B Counters, so it must count a number of clocks which have a common denominator to 665 and 945. For ease, I have set the Master Counter to count 5 clock periods before sending out that pulse, so $\&(16) = 4$ (0,1,2,3,4 = 5) Now that means Counter A must be set to $665/5=133$, therefore $\&(17)=132$. And Counter B must be set at $945/5=189$, therefore $\&(18)=188$.

If you run these results back through the first formulas, you will find that Freq A = 1336.8 hertz and Freq B = 940.7 hertz, well within the tolerance band.

To have both voices come from the tv speaker at the same time only requires that Volume A and Volume B be brought up to sufficient numbers. Both would be maximum if $\&(22)$ is set to 255.

I have found two exceptions that may be unique to my own computer... One, when $\&(16)$ is set to 0, it recognizes it as 2; and the other is when $\&(16)$ is set to 1, it recognizes it as 3. I can't explain why, it may have to do with the high speed at which you are trying to make the Master Counter work.

Load the "SOUND GRAPH" program and get some 'hands on' experience, and try out different ideas. Also, there is an optional modification to the program that will display all eight binary bits of the value that the knob is adjusted to at that time. It does slow down the program, but it is a good teaching aid.

PROGRAM NAME SOUND GRAPH

Page 1 of 1

Comments

Statements(s)

Line #

```

4 :RETURN
5 :SOUND GRAPH
6 CHUCK THOMKA 28AP79
10 CLEAR:NT=0
20 FOR A=16 TO 23
30 PRINT #1,A;NEXT A
40 M=KN(1)+1.28
50 IF M=Y GOTO 90
60 V=M
80 Y=-32:GOSUB 200
90 L=L-JY(1)
100 IF L<11=1
110 IF L>8L=8
120 Y=48-8*L
130 BOX -74,Y,12,7,2
140 CX=-77:CY=Y
150 PRINT #1,15+L
160 IF TR(1) GOSUB 200:R(1)
5*L)=Y
170 GOTO 40
200 BOX 6,Y,148,7,2
210 IF V+2=0 GOTO 230
220 BOX Y+4-68,Y,Y+2,3,1
230 CX=64:CY=Y
240 PRINT #1,Y;RETURN

THIS IS OPTIONAL
20 GOSUB 300
300 CY=-40:B=Y
310 FOR F=10 TO 3 STEP -1
320 CX=-77
330 B=B+2
340 PRINT #F,RM,
350 NEXT F
360 RETURN

```

PROGRAM NAME TOUCH TONE SIMULATE

Page 1 of 1

Comments

Statements(s)

Line #

```

4 :RETURN
5 :TOUCH TONE SIMULATE
6 CHUCK THOMKA 28AP79
10 CLEAR
20 A=0
30 NT=3:N=KP
40 IF N=11.6 GOTO 110
50 IF N=105 GOTO 100
60 IF N=31A=A-1:GOTO 90
70 A=A+1
80 Q(A)=N
90 TV=N
100 GOTO 30
110 NT=0:PRINT
120 FOR P=1 TO A
130 TV=Q(P)
140 GOSUB 252+Q(P)
150 R(17)=C:R(18)=R
160 R(16)=4:R(22)=2.55
170 FOR T=1 TO 75
180 NEXT T
190 R(22)=0
200 NEXT P
210 GOTO 30
300 C=132:R=188:RETURN
301 C=146:R=254:RETURN
302 C=152:R=254:RETURN
303 C=119:R=254:RETURN
304 C=146:R=230:RETURN
305 C=132:R=230:RETURN
306 C=119:R=230:RETURN
307 C=146:R=208:RETURN
308 C=132:R=208:RETURN
309 C=119:R=208:RETURN

```

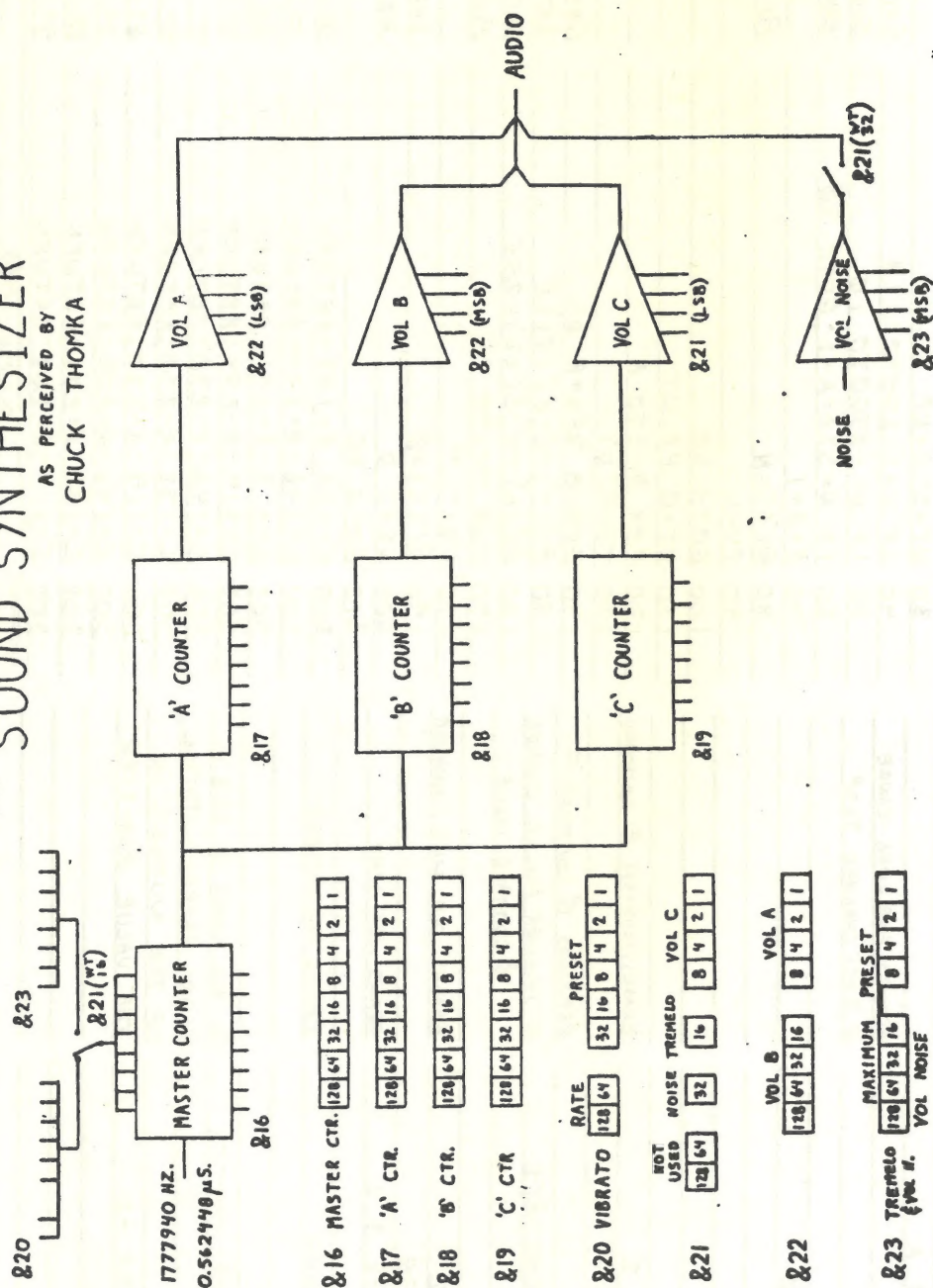
COLUMN FREQS

| BELL | 1209 | 1336 | 1477 | 1633 | 1789 |
|----------|------|------|------|------|------|
| SYSTEMS | 1 | 2 | 3 | 4 | 5 |
| REQUIRED | 4 | 5 | 6 | 7 | 8 |
| FREQ. | 7 | 8 | 9 | 10 | 11 |
| GAID | * | 0 | # | 0 | 941 |

Comments

COMMANDS ARE "PRINT", "CLEAR" AND "ERASE". THESE WILL
 PLAY NUMBERS IN MEMORY
 CLEAR MEMORY
 ERASE ONE NUMBER
 PUT NUMBER IN MEMORY
 GET FREQ FM TABLE
 SET IN NEW FREQS
 SET MASTER OSC & VOLUME
 PLAY NUMBER LOOP
 SHUT OFF SOUND
 NEXT NUMBER TO DIAL
 ONE COLUMN FREQ AND ONE
 ROW FREQ PER NUMBER
 PUSHED. ALSO I'VE ADDED
 A TO D IF YOU HAVE THE
 16 KEY KEY-PAD PHONE

THE SOUND SYNTHESIZER
AS PERCEIVED BY
CHUCK THOMKA



PROGRAM NAME MEMORY DISPLAY

Line # Statement(s)

```

4 :RETURN
5 MEMORY DISPLAY
6 CHUCK THOMKA 4APR79
10 INPUT "FIRST LOCATION"
  #="F
20 INPUT "LAST LOCATION"
  #="L
30 INPUT "STEP AMOUNT ="
  S
40 FOR N=FTO L STEP S
50 PRINT #6,N," ",#1," ",#2," "
60 E=N;GOSUB 200
70 H=#2(N)
80 PRINT #1," "=(#1)
90 E=H;GOSUB 200
100 PRINT #1," "),(#1),#6,H
110 NEXT N
120 GOTO 10
200 S=0
210 IF E<0 E=E+32767+1;S=S+8
220 E=E÷16;D=RM
230 E=E÷16;C=RM
240 E=E÷16;B=RM
250 A=E+S
260 E=A;GOSUB 400
270 E=B;GOSUB 400
280 E=C;GOSUB 400
290 E=D;GOSUB 400
300 RETURN
400 IF 0<E IF E<10 PRINT
  #1,E;RETURN
410 TV=55+E;RETURN
  
```

Comments

THIS IS A NICE MEMORY DUMP
PROGRAM THAT DISPLAYS
THE DECIMAL AND HEXADECI-
MAL LOCATION NUMBERS
(ADDRESS) AND DATA. IT
WILL DO WHOLE BLOCKS
OF DUMPS BY GIVING A
STARTING AND ENDING
ADDRESS. IT WILL INCRE-
MENT THE ADDRESS BY THE
ENTERED AMOUNT IF YOU
ONLY WANT TO CHECK
EVERY 1000th LOCATION.
FOR EXAMPLE,

Page 1 of 1

Line # Statement(s)

```

1 >SQUARE ROOTS
2 BY DAVID STOCKER
10 INPUT S
20 A = S÷2
30 FOR T=1 TO 10
40 IF A=0 A=1
50 B=S÷A
60 IF A=B GOTO 100
70 A=(A+B)÷2
80 NEXT T
100 PRINT A,B (A WILL = B ± 1 (ERROR)
110 PRINT A,B
120 PRINT A-B-S (OPTIONAL)
130 PRINT S
  
```

Comments

USE OF SPACE IN A IS FOR TWO OR
MORE LINES OF MULTI-LINE STATEMENTS

>DISTANCE BETWEEN

TWO POINTS

BY DAVID STOCKER

CLEAR;PRINT;PRINT

INPUT X } POINT 1

INPUT Y } POINT 2

INPUT "X" V

INPUT "Y" W

BOX X₁,Y₁,X₂,Y₂

BOX X₁,Y₁,X₂,Y₂

P=(X-Y) * (X-Y) + (Y-W) * (Y-W)

A=P÷2

FOR T=1 TO 10

IF A=0 A=1

B=P÷A

IF B=A GOTO 200

A=(A+B)÷2

NEXT T

PRINT A,B (A WILL = B ± 1 (ERROR)

DO NOT ENTER A SPACE BETWEEN LINE #
AND STATEMENT, THIS IS DONE BY THE UNIT

ADS

- o Bally Basic cartridge, manual, interface package, SEAWOLF/MISSILE, BRICKYARD/CLOWNS for \$80. M. Jones 11106 Polaris San Diego, CA 92126 714-566-8087
- o CHESS for two players \$6 on my tape or \$5 on yours. Add \$2 for CHECKERS or HANGMAN on reverse. J. Collins 713 Bradford Dr. Ft. Walton Beach FL 32548
- o Program listings available from Ron Schwenk, 6988 Lincoln Creek Circle, Carmichael CA 95608 - Mastermind/Batnum at .20 each, Craps at .10, Legal stamped envelope, please
- o SEA BATTLE @4./HORSE RACE, STARSHIP, ROBOT WAR @3.each/STAR WARS, SLOT MACHINE, STARTREK, CRAPS 2, CONNECT 4 @2.each/TICTACTOE @1. Above on your tape, 4 min per program, half price for listing alone. All ten for \$22. All except Connect 4 have graphics. Scott Waldinger 24740 Woodcroft Dr. Dearborn MI 48124

BOOKS that have been recommended by subscribers are: A GUIDED TOUR OF COMPUTER PROGRAMMING IN BASIC by Dwyer and Kaufman, and 57 PRACTICAL PROGRAMS & GAMES IN BASIC by Tracton. I have received a couple of programs that were modified from the latter.

SERVICES AVAILABLE from Tim Hays, 456 Granite Ave, Monrovia CA 91016 include: piano-like 3-octave keyboard; hard wired TBASIC chip into motherboard with switch; Sound output; special application software, games, music, sound effects, also sales of Bally hardware. Send for details 213-359-8092

VARIETY is offered by W&W Software Sales, 6594 Swartout Rd. Algonac MI 48001. Five pre-programmed tapes with 5 programs each at \$10 each tape. Guaranteed bug-free. Include boards for games, colors, skill levels, business programs. Send S.A.E.

THIS ISSUE is pretty well packed with stuff, primarily your contributions of one kind or another. I have received a lot of compliments on the paper, but I am only relaying the material that comes in to me, so the compliments really go right back out to you, the subscribers. On the other hand, "keep those cards and letters comin' ", so that we can all benefit in this endeavor.

REWORKED GAMES mentioned above also have been taken from 24 TESTED, READY-TO-RUN GAME PROGRAMS IN BASIC by Tracton (The Tracton books are distributed by TAB books) and from BASIC COMPUTER GAMES by Ahl, editor of Creative Computing magazine. A recommended book is THE BASIC COOKBOOK by Tracton, a dictionary type of book. These have been reported by Bob Kelley.

68

ARCADIAN

Robert Fabris, folder
3626 Morrie Dr.
San José, CA 95127

FIRST CLASS